SPArTaCuS: Service Priority Adaptiveness for Emergency Traffic in Smart Cities using Software-Defined Networking

Rohit Abhishek, Shuai Zhao, Deep Medhi
Computer Science & Electrical Engineering Department
University of Missouri–Kansas City, USA
{rabhishek, shuai.zhao, dmedhi}@umkc.edu

***Abstract*—Network congestion during disasters and big events is a major issue, especially in metropolitan areas. Although different network operators have their own strategies to address such types of incidents, a smarter and more efficient way to address such situations is needed for Smart Cities. In this paper, we propose SPArTaCuS, a framework to prioritize network traffic adaptively for such situations for smart cities using a software-defined network (SDN) approach, where services that require priority are placed in virtualized networks and the mechanism is accomplished through a priority management layer in SDN architecture.**

I. Introduction

Network congestion is a major issue during disasters and sporting or big events when the network is highly utilized, and it is not able to carry any requests as the demand goes beyond the capacity of the network. When such situations happen, Mobile Network Operators (MNOs) face challenges to ensure uninterrupted network operations and to specifically give priority to the governmental communication services for relief efforts [1]. During such situations, the operators are required to prioritize calls and to manage other demands in their network or do on-demand spectrum allocation for emergency service in a stressed situation [2]. The disruption of communication services during the disaster can hamper the relief efforts, thus creating more chaos.

During highly popular sporting events or big events such as concerts by popular artists, the network gets congested due to a large number of users attempting to call, streaming videos or uploading photos. These activities congest the network and affects people who are trying to reach emergency services during such events. For example, in February 2014, during the Seahawks Super Bowl parade, the Seattle Emergency Operations Center sent an alert that asked people to wean off their cell phone use to keep 911 networks open [3]. Again, at the Torchlight Parade in July, Seattle Police asked citizens to text friends and family instead of calling to prevent the network from getting congested [3]. A similar situation was experienced in November 2015 during the Kansas City Royals' World Series Champion celebration parade attended by reportedly 800,000 fans. Many people could not tweet, text, or make calls as the network was overwhelmed causing a cellular traffic jam [4]. Although the network operators were somewhat prepared for it ahead of time, they still experienced unusually heavy congestion because too many fans showed up at this event. Typically, for a stadium with a capacity of less than 100,000 people, the wireless operators often temporarily add extra capacity by deploying *Cell on Wheels* (COW) [5] or airborne base stations [6]; furthermore, many sports stadiums are now deploying wi-fi services for fans to post to social media. However, planning for a one time event for 800,000 people is almost impossible to handle as the load is extremely high[1]; yet, emergency services still should get network access during such an event.

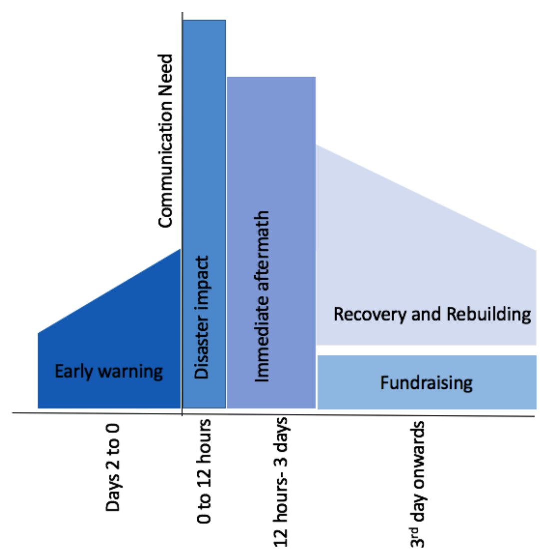

Fig. 1. Communication needs at different phases of a disaster [7]

In the case of known disasters such as hurricanes, the graph in Fig. 1 shows the communication needs from two days prior to the disaster till three days after the disaster [7]. We can see that how the communication need increases during the early warning signs of a disaster and the need for

[1]One of the authors was at the event and could not get any connectivity for either phone or data services for nearly two hours on his smartphone.

978-1-5090-1846-8/16/$31.00 ©2016 IEEE

communication is greatest during the impact of the disaster as well as immediately afterwards as this is the time when everyone is calling their loved ones and all the calls to the helpline services are made.

In this work, we propose SPArTaCuS (Service Priority Adaptiveness for Emergency Traffic in Smart Cities using Software-defined networking), a framework for smart cities on how to prioritize services for emergency needs in a stressed situation. Our approach is based on a promising new networking technology, known as software-defined networking (SDN). Our approach resorts to virtualizing networks for different service classes and dynamic allocation of resources as needed.

The rest of the paper is organized as follows. In Section II, we first review SDN followed by our proposed architectural framework in Section III that gives the architecture overview, the use of middlebox and priorities for different Virtual Networks (VNs) during different scenarios. Section IV states the related work, and Section V gives the conclusion and the future work.

II. SDN: Overview

SPArTaCuS relies on the underlying network function being provided by SDN. We start with a brief overview of SDN and the features available with its latest implementations that will be utilized in SPArTaCuS. Briefly, SDN is a networking architecture that allows dynamic, manageable, cost-effective, and adaptable services, making it ideal for high-bandwidth in the dynamic nature of today's applications [8]. Fig. 2 gives an overview of SDN. The control is centralized in software-based SDN controllers giving it the global view of the network due to which the network is viewed as a single, logical switch to the applications and policy engine [9]. The control plane and the data plane are separated. Some of the major features of SDN architecture are [8] as follows:(i) the network control can be directly programmed; (ii) the network can be dynamically adjusted to network-wide traffic flows; (iii) the controller maintains a global view of the network; (iv) it lets network managers configure, manage, secure, and optimize network resources efficiently via dynamic, automated SDN programs; (v) it simplifies network design and operation, because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

The SDN controller is the application in the SDN network that is responsible for managing flow control to the data plane on the southbound interface and the applications on the northbound interface using API calls [10] (Fig. 2). OpenFlow [11] is one popular southbound API for an SDN network. Some of the common SDN controllers' implementations are RYU [12], POX [13], OpenDaylight [14], and Floodlight [15].

OpenVirtex (OVX) is a network virtualization platform that gives the facility to create and manage virtual Software-Defined Networks [16]. It can create multiple virtual and programmable networks on top of a single physical infrastructure. In the OpenVirtex Architecture (Fig. 3), the physical layer is mapped to OpenVirtex, where the physical topology is virtualized and this virtual topology is connected to the SDN

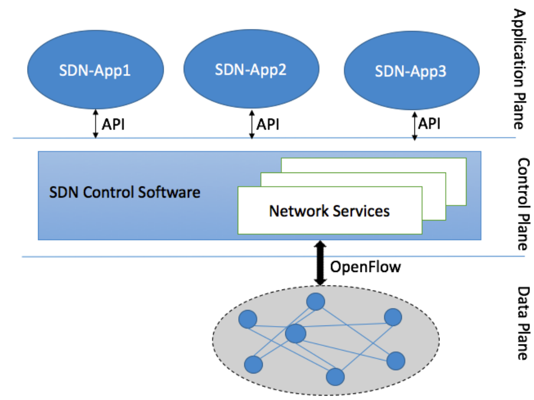

Fig. 2. Software-Defined Network Architecture

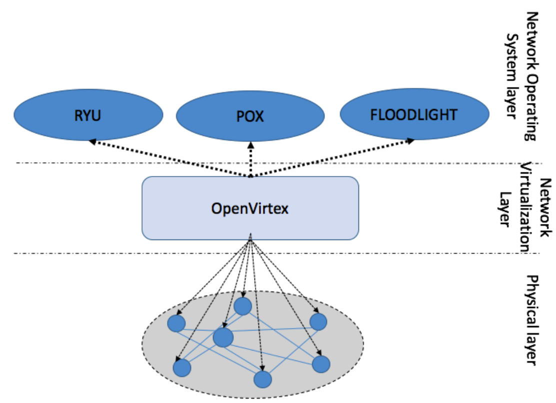

Fig. 3. OpenVirtex Framework

controller, allowing multiple SDN controllers to be connected to OpenVirtex at the same time.

A few essential features provided by OpenVirtex are [17] as follows: (i) topology customization to customize the virtual SDN topologies that are visible to each controller but are isomorphic to the infrastructure, (ii) resilience so that a virtual link or switch can be mapped onto multiple physical components to provide redundancy, (iii) topology isolation meaning the virtual networks (VNs) do not know the existence of each other, (iv) resources can be allocated and de-allocated dynamically, (vi) different controllers can be use to provide resilient services.

III. SPArTaCuS

Our proposed approach (SPArTaCuS) uses SDN to accomplish service prioritization for emergency services in a stressed situation. In particular, SPArTaCuS uses the SDN framework with OVX to create virtual SDN networks for different service classes that are mapped to the physical infrastructure. Fig. 4 presents a high-level view of the SPArTaCuS framework. In our approach, the middlebox layer has a priority management layer on top of OVX that is connected to multiple SDN controllers on the northbound interface.

Fig. 5 shows internal architecture with virtual networks in SPArTaCuS. The middlebox layer is used to create the VNs

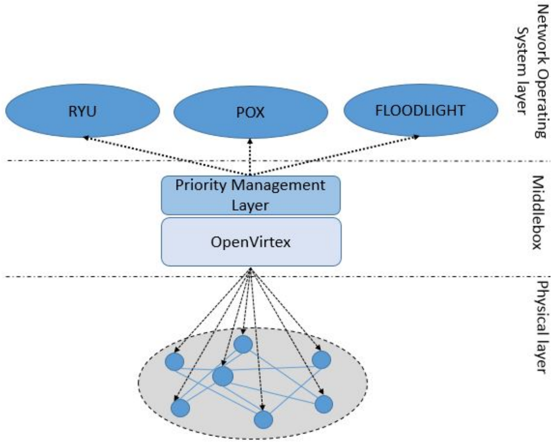

Fig. 4. SPArTaCuS: Architecture Framework

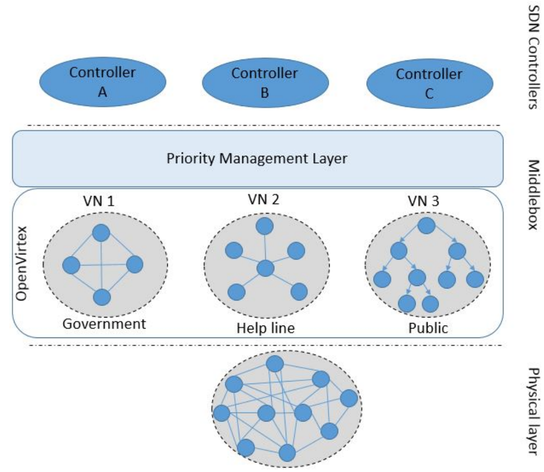

Fig. 5. Modelling Smart Cities networks in SPArTaCuS

and provide priority to them. The OVX creates the VNs, whereas prioritization is done by the priority management layer. In the middlebox, virtual networks are created for different entities or organizations. For example, traffic for government communication can be directed via a specific set of VNs that are responsible for government networks. Similarly, we can classify other virtual networks according to different traffic such as helplines and for the general public. We illustrate three VNs in SPArTaCuS shown as VN1, VN2 and VN3; here, VN1 is responsible for government traffic, VN2 for helpline services, and VN3 is categorized for public traffic.

A. Architectural Framework

Resource allocations to each of the VNs are done by the priority management layer. These allocations are based on the ratio of traffic for various organizations. The priority management layer decides resources that are to be de-allocated or added to the different VNs based on a policy on priority. Here, priority can be set up manually or dynamically for different scenarios. The priority management layer is responsible for resource de-allocation and allocation. Resources for each VN are allocated by adding nodes and links to the existing VNs, while resource de-allocation is done by removing nodes and the associate links. The priority management layer decides which node to de-allocate by finding the edge nodes. If any core node is de-allocated, then the majority of the network maybe down. Similarly, to decide where to add nodes is based on finding the optimal point in the network where these nodes can help in the highest demand flows.

As shown in the Fig. 4, the physical network is connected to the middlebox layer through OVX. Each SDN controller can serve one VN or one SDN controller can server multiple VNs. However, different VNs work in parallel and are isolated from each other.

B. Situation Aware Invocation

We now discuss two potential situations where network prioritization can be invoked dynamically.

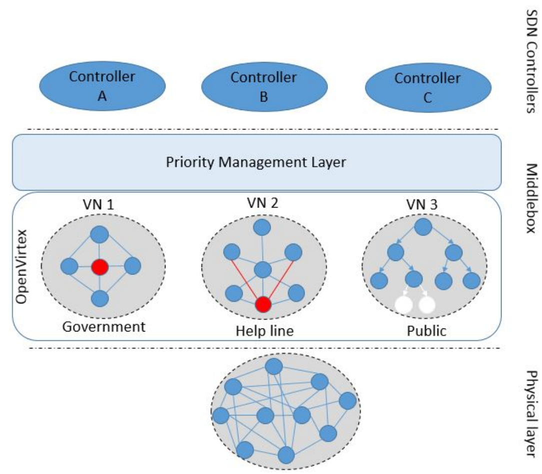

Fig. 6. Network prioritization during disaster (VN2≻VN1≻VN3)

Natural Disaster: During natural disasters such as hurricanes and earthquakes, both helpline services and public traffic are the most utilized ones. Keeping in mind the number of casualties, our approach prioritizes traffic destined for helpline services. The priority management layer takes away some resources from VN3 that is responsible for public and instead allocates them to VN2 that is responsible for helplines. Also, some resources can be allocated to VN1, which is responsible for the government services' traffic. So here, the first priority will be helplines, then the government and then the public. We denote this priority policy as: VN2≻VN1≻VN3. In Fig. 6, we show that some resources have been de-allocated from VN3 and are added to VN2 and VN1; here, red nodes and links are the ones that have been added to VN1 and VN2 by de-allocating resources from VN3 marked in white nodes and links.

Sporting Event or Big Events: Consumer demands for wireless technology have risen because of the popularity of smart phones where users tend to use different apps at any instant [18]. This means that they are constantly using some

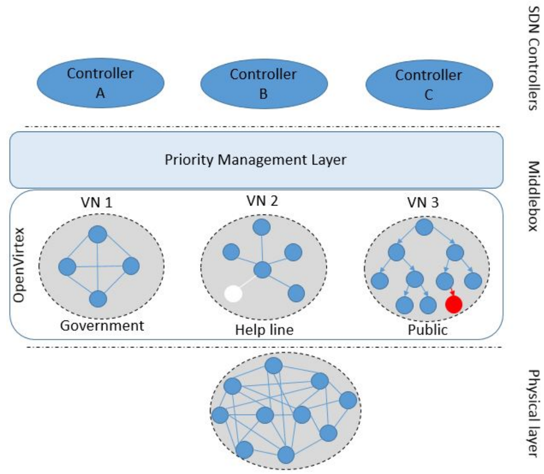

Fig. 7. Network prioritization during big events (VN3≻VN2≻VN1)

resources at any given time. During sporting events or any big events, call signals might break and eventually drop — this is a bandwidth problem that arises due to having a large number of users in a small geographic area resulting in network congestion. For such situations, Cell on Wheels (COW) is commonly deployed. However, COW may have shortcomings so it may not be able to fulfill the requirement, and in some cases, it may interfere with neighboring cell towers that eventually would degrade the service.

Our idea is to prioritize the network that is being used to route the public traffic. The Priority Management Layer will prioritize the traffic by allocating more resources to the VN3. Thus, we can de-allocate some resources from VN2 and allocate them to VN3. It is important to keep in mind that resources de-allocated from a VN have no adverse effect on these VNs. For example, it is important to ensure that we de-allocate resources from VN2, which is responsible for the helpline, in such a way that it does not affect normal traffic flow destined for the helpline services. Fig. 7 shows the network after the network priority is done for the public VN3. Nodes and links marked in white in VN2, denote resources that have been de-allocated while the red nodes and links in VN3 denote the resources that are being added to VN3. The priority can be stated as VN3≻VN2≻VN1.

The middlebox that consists of OVX and the priority management layer, creates different VNs that are prioritized based on the scenarios, like disasters and sporting or big events, where the network gets congested.

IV. Related Work

There have been a number of works on how to handle the network during disasters and big events [1]–[3], [18]. [2] aims at characterizing the network infrastructure behavior during natural disasters by studying the scenarios from a device level prospective. Here, they studied two disaster scenarios and characterized the network infrastructure. [19] is a Disaster Response report on how AT&T handles natural disasters, while [1] addresses the technical challenges that mobile operators face during disasters.

All of these works have been developed for traditional IP networks, not for SDN. We have proposed an SDN-based approach to handle traffic prioritization in stressed situations using virtual SDNs.

V. Conclusion and Future Work

In this paper, we propose SPArTaCuS to prioritize traffic based on SDNs for different service classes. We argue to divide the traffic based on different organizations and prioritize them using the priority management layer in the middlebox. We illustrate two situations where SPArTaCuS can be helpful. In the future, we plan to implement a SPArTaCuS prototype and also develop (in detail) specific allocation/de-allocation schemes based on policy-level directives and service level agreements.

Acknowledgement

This work is partially supported by NSF Grant # CNS-1217736.

References

[1] GSMA, “Dealing with disasters: Technical challenges for mobile operators,” 2012. http://www.gsma.com/mobilefordevelopment/wp-content/uploads/2012/06/Dealing-with-Disasters_Final.pdf.
[2] Z. Gilani, A. Sathiaseelan, J. Crowcroft, and V. Pejovi, “Inferring network infrastructural behaviour during disasters,” in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2016, pp. 642–645.
[3] “http://www.geekwire.com/2014/city-seattle-emergency-cell-phone/.”
[4] J. R. Eubank, “First it required a disaster, now all it takes is a ‘parade’,” november 10, 2015. https://www.linkedin.com/pulse/first-required-disaster-now-all-takes-parade-john-r-eubank.
[5] “Cell on wheels (COW),” http://www.icsindustries.com.au/products/communication-trailers/cell-on-wheels.
[6] A. Valcarce, T. Rasheed, K. Gomez, S. Kandeepan, L. Reynaud, R. Hermenier, A. Munari, M. Mohorcic, M. Smolnikar, and I. Bucaille, “Airborne base stations for emergency and temporary events,” http://arxiv.org/pdf/1307.3158.pdf.
[7] GSMA, “The role of mobiles in disasters and emergencies,” dec 2005. [Online]. Available: http://www.enlightenmenteconomics.com/about-diane/assets/disasterreport.pdf
[8] “Software-defined networking (SDN) definition.”
[9] Open Networking Fundation, “Software-defined networking: The new norm for networks,” *ONF White Paper*, apr 2012, https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.
[10] “https://www.sdxcentral.com/resources/sdn/sdn-controllers/.”
[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks (openflow white paper),” *http://www.openflowswitch.org*, 2008.
[12] “Ryu SDN framework,” https://osrg.github.io/ryu/.
[13] “POX,” http://www.noxrepo.org/pox/about-pox/.
[14] “OpenDaylight platform,” https://www.opendaylight.org/.
[15] “Project Floodlight,” http://www.projectfloodlight.org/floodlight/.
[16] “OpenVerteX,” http://ovx.onlab.us.
[17] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, G. Parulkar, E. Salvadori, and B. Snow, “OpenVirteX: Make your virtual SDNs programmable,” in *Proceedings of the third workshop on Hot topics in software defined networking*. ACM, 2014, pp. 25–30.
[18] “Wireless coverage at stadiums and other high traffic venues: Visitor satisfaction and owner value,” http://www.tpidas.com/pdf/DAS.pdf.
[19] F. McGrath, “Disaster response – business as usual: How AT&T deals with natural disasters,” dec 2014, http://www.gsma.com/mobilefordevelopment/wp-content/uploads/2015/01/GSMA_Business-as-Usual-How-ATT-Deals-with-Natural-Disasters.pdf.